

思考题

1. 为何在数据增强前要加入 `if usage == 'train'` 的判断式？

答：数据增强的目的是提供模型更多不同的训练数据，评估是不需要做数据增强。

2. 有哪些数据增强的方法（举例）？

答：随机剪裁、随机翻转、随机旋转等

3. 什么是一个 step？

答：一个 step 指一个 batch 送入网络中完成一次前向计算及反向传播的过程。

4. 编写自定义回调函数需继承什么基类？

答：Callback 基类

5. 如何在自定义回调函数中获取训练过程中的重要信息（损失函数、优化器、当前的 epoch 数等）？

答：通过 `run_context.original_args()` 方法可以获取到 `cb_params` 字典，字典里会包含训练过程中的重要信息。

6. 本实验的模型欠拟合，可通过增加网络复杂度优化模型。如果模型过拟合，常见的处理方法有哪些？

答：L1, L2 正则化, Early stopping, 增加数据集, 增加噪声, 添加 dropout 层等。

7. 如何理解 dataset sink mode？

`dataset_sink_mode=True` 时，可以这样简单的理解：Model 中仅建立数据通道与执行网络之间的连接关系，不会直接将数据喂给网络。数据会通过数据通道(pipeline)下发到卡上，网络在卡上执行时会直接从对应的数据通道中获取数据。在这种模式下，数据下发与网络执行可以并行，单个 epoch 的训练过程中 host 与 device 之间不会交互，因此能提升性能，也因此打屏以 epoch 递增。

`dataset_sink_mode=False`，没有使用数据下沉模式。这时候数据不会通过通道直接向 Device 下发，Model 会将数据一个 batch 一个 batch 得取出，喂给网络。在这种模式下，每个 step 结束，host 都可以获取 device 上网络的执行结果，因此打屏以 step 递增。

8. 为什么本实验的模型在训练集的 loss 波动较大，且比在测试集高？

答：因为训练集做了数据增强，有些经过剪裁或翻转的图片较难辨认。

优化前：

载入最佳网络参数，并测试其 accuracy 与 loss

```
In [40]: # 使用准确率最高的参数组合建立模型，并测试其在验证集上的效果
load_checkpoint(os.path.join(ckpt_path, 'best_param.ckpt'), net=network1)
res = model.eval(test_data, dataset_sink_mode=dataset_sink_mode)
print(res)

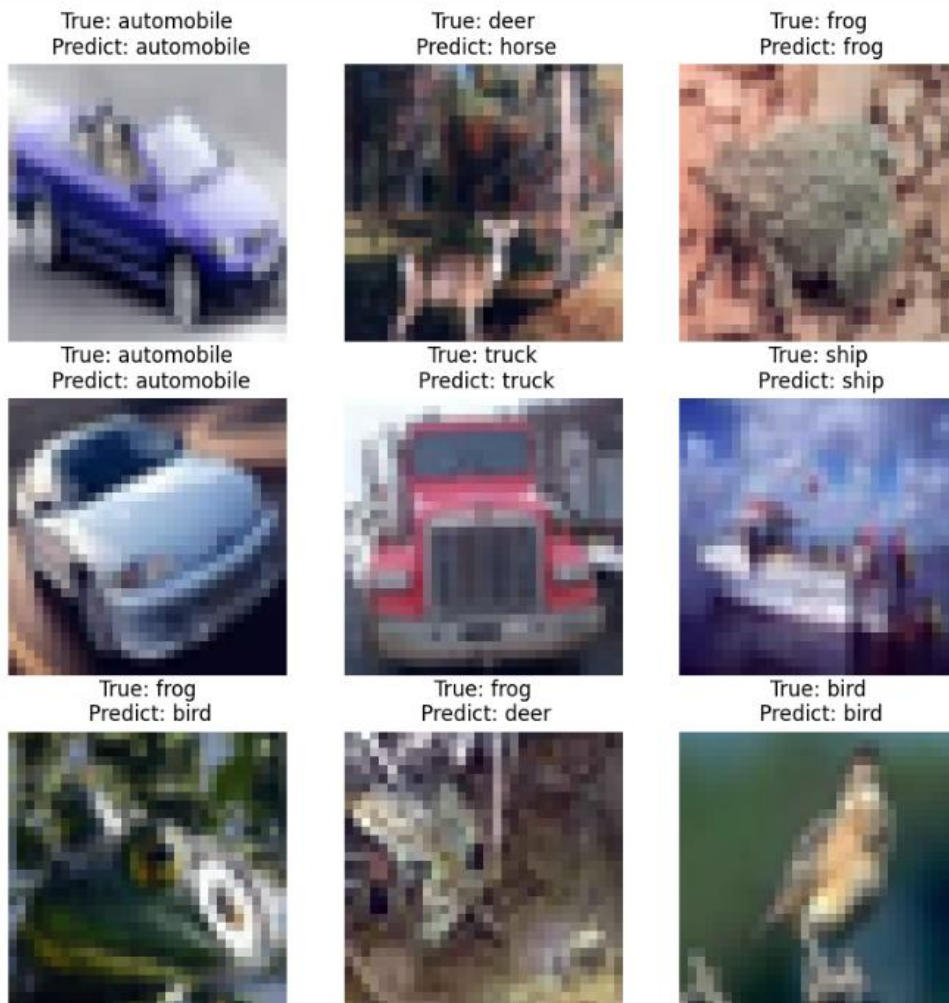
{'accuracy': 0.5716, 'loss': 1.225158831179142}
```

优化后：

载入最佳网络参数，并测试其accuracy与loss

```
in [48]: # 使用准确率最高的参数组合建立模型，并测试其在验证集上的效果
best_param = mindspore.load_checkpoint(os.path.join(ckpt_path, 'best_param.ckpt'), net=network2)
res = model.eval(test_data, dataset_sink_mode=dataset_sink_mode)
print(res)

{'accuracy': 0.7378, 'loss': 0.7650443188846111}
```



googlenet 优化:

```
lectKernel] There are 70 node/nodes used reduce precision to selected the kernel!
accuracy: {'acc': 0.9348958333333334}
```